



Übung Python Netzwerkprogrammierung UDP/TCP

Sofern nicht anders angegeben sind sämtliche Aufgaben in der Programmiersprache Python umzusetzen. Erstellen Sie dazu ein eigenes Virtual Environment.

Aufgabe 1:

Erstellen Sie einen UDP Client welcher Daten an eine zu definierende IP-Adresse/Portnummer sendet.

Erstellen Sie einen UDP Server welcher an einem bestimmten Port und an einem zu definierenden Netzwerkinterface auf eingehende Daten wartet.

Portnummer/IP-Adresse/Netzwerkinterface sollen einfach geändert werden können.

Der Client soll alle 1ms ein Datenpaket an den Server senden können.

Der Server soll für jedes 1000. Datenpaket bestimmte Daten ausgeben. Also pro Sekunde eine Ausgabe.

Als Daten senden sie genau einen INT32 Wert. Also in Summe pro Datenpaket 4 Byte. Der gesendete Integerwert soll erst mal 0 sein.

Für die Umwandlung von Zahlen in Binary Daten verwenden sie das Python package „struct“.

<https://docs.python.org/3/library/struct.html>

```
1  import struct
2
3  # Liste mit 2 Integer Zahlen erzeugen
4  data = []
5  data.append(1)
6  data.append(2)
7
8  # es sollen 2 Integerzahlen (INT32) in Binärdaten umgewandelt werden
● 9  xx = struct.pack("!2i", *data)
10
11 # Binärdaten in 2 Integerzahlen zurückwandeln
12 returnData = struct.unpack("!2i", xx)
```

Prüfen Sie ob sie Daten zwischen Client und Server austauschen können.

**Aufgabe 2:**

Erweitern Sie die Client und Server dahingehend das sie eine bestimmte Anzahl an INT32 Werten in einem Datenpaket senden/empfangen können. Die Anzahl der INT32 Werte soll einstellbar sein. Im ersten INT32-Wert soll eine fortlaufende Nummer übertragen werden. Beginnend mit 1 beim 1. Datenpaket bis n beim n-ten Datenpaket. Alle anderen INT32 Werte des Pakets können auf 0 gesetzt werden.

Beispiel von 3 Datenpaketen. In jedem Paket werden 3 INT32 Werte übertragen.

```
1. Datenpaket: 00|00|00|01|00|00|00|00|00|00|00|00
2. Datenpaket: 00|00|00|02|00|00|00|00|00|00|00|00
3. Datenpaket: 00|00|00|03|00|00|00|00|00|00|00|00
```

Die ersten 4 Byte sind für den ersten INT32 Wert. Die nächsten 4 Byte für den nächsten INT32 Wert usw.

Der Server welcher die Daten empfängt soll den Binären Datenstrom wieder in INT32 Werte zurückwandeln und den ersten INT32 Wert für jedes Datenpaket entsprechend auswerten. Hier steht ja die fortlaufende Nummer.

Wenn der Server nun die Anzahl an empfangenen Datenpaketen mitzählt und diese immer mit dem Wert des ersten INT32 Wert pro Datenpaket vergleicht kann er feststellen ob Pakete verloren gegangen sind.

Nach jedem 1000. Datenpaket soll der Serverprozess ausgeben wieviele Datenpakete empfangen wurden und wieviele verloren gegangen sind (Absolut und in Prozent angeben).

**Aufgabe 3:**

Erstellen Sie das gleiche Programm wie in Aufgabe 2. Allerdings mit TCP als Transportprotokoll anstatt UDP.

Der Server soll permanent auf eingehende Verbindungen „horchen“. Baut ein Client die Verbindung auf, so soll ein eigener Thread gestartet werden, der für den Datenaustausch zw. Server und genau diesem Client zuständig ist.

Verbinden sich also z..b. 3 Clients, so müssen 3 Threads gestartet werden.

Finden Sie auch heraus wie sich ein „Verbindungsende“ auf den Client/Server auswirkt und gehen sie im Programm entsprechend damit um.

```
1  import threading
2  import time
3
4  threadRunning = True
5  def ThreadFunction(data1, data2):
6      global threadRunning
7      while threadRunning:
8          print (f"{data1} - {data2}")
9          data2 += 1
10         time.sleep(1)
11
12     print (f"Thread finish: {data1}")
13
14 # Start 2 Threads. Same thread function. Different data
15 th = threading.Thread(target=ThreadFunction, args=("Thread1", 17))
16 th.start()
17
18 th = threading.Thread(target=ThreadFunction, args=("Thread2", 100))
19 th.start()
20
21 # Main Loop
22 try:
23     while True:
24         time.sleep(1)
25 except KeyboardInterrupt:
26     threadRunning = False
27     print ("finish")
```

Aufgabe 4:

Führen Sie nun Messungen durch (mit der TCP und der UDP Variante). Client und Server müssen dazu auf 2 versch. Computern gestartet werden.

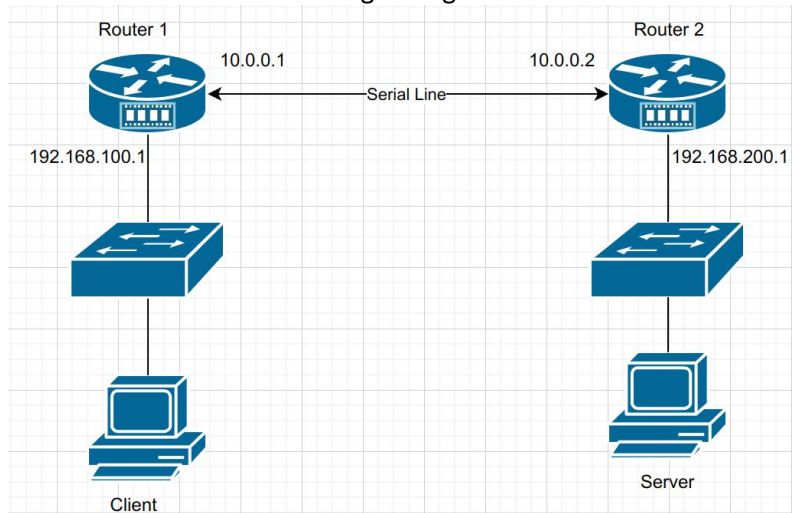
Beide PCs sind über ein LAN Kabel, Switch, Schulnetz verbunden.

Beide PCs kommunizieren über das Schul-WLAN miteinander.

Für die Messungen lassen sie jeweils Client/Server 2-3 Minuten laufen.

Aufgabe 5:

Bauen sie nun mit den Rollwägen folgendes Testnetzwerk auf.



Client und Server sollen Ihre IP-Adresse automatisch über DHCP beziehen.

Erstellen Sie für Router1 und Router2 entsprechend eine Konfiguration sodaß Client und Server miteinander kommunizieren können. Als Routingprotokoll verwenden sie RIP.

Testen Sie das Netzwerk ob Client und Server sich gegenseitig pingen können.

Starten sie nun den UDP-Client und UDP-Server mit einer Paketlänge von 5 INT32-Werten (20 Byte). Hierbei ist zu erwarten das es zu keinen Paketverlusten kommt.

Überlegen Sie nun wie groß die Paketlänge werden darf bis erste Paketverluste auftreten. Begründen Sie Ihre Antwort. Hinweis wir senden alle 1ms ein Datenpaket mit entsprechender Länge.

Erhöhen Sie nun die Paketlänge und überprüfen Sie Ihre Annahme.

Hinweis:

Sämtliche Übungen sind in einem entsprechenden Protokoll zu dokumentieren. Der Quellcode inkl. kurzer Erklärung, sowie die Routerkonfigurationen sind anzugeben. Die Ergebnisse z.b.mit Screenshots zu dokumentieren.